



Calhoun: The NPS Institutional Archive DSpace Repository

Theses and Dissertations

1. Thesis and Dissertation Collection, all items

1995-06

Graphical user interface network applications

Lanni, M. J.

Monterey, California. Naval Postgraduate School

<http://hdl.handle.net/10945/31581>

This publication is a work of the U.S. Government as defined in Title 17, United States Code, Section 101. Copyright protection is not available for this work in the United States.

Downloaded from NPS Archive: Calhoun



<http://www.nps.edu/library>

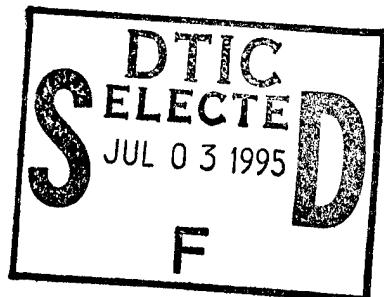
Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community.

Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

**Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943**

NAVAL POSTGRADUATE SCHOOL

Monterey, California



THESIS

GRAPHICAL USER INTERFACE NETWORK APPLICATIONS

by

M. J. Lanni

March 1995

Thesis Advisor:

Norman F. Schneidewind

Approved for public release; distribution is unlimited

19950629 017

DTIC QUALITY INSPECTED 5

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY (<i>Leave blank</i>)	2. REPORT DATE March 1995	3. REPORT TYPE AND DATES COVERED Master's Thesis	
4. TITLE AND SUBTITLE Graphical User Interface Network Applications		5. FUNDING NUMBERS	
6. AUTHOR(S) M. J. Lanni			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey CA 93943-5000		8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited		12b. DISTRIBUTION CODE	
13. ABSTRACT (<i>maximum 200 words</i>) This thesis presents a research study of configurations for installation of graphical user interface and software applications on a local area network. Requirements for a graphical user interface on a local area network are discussed, specifically the Systems Management Department Laboratories at Naval Postgraduate School. Compatibility and interoperability of network operating system, PC operating system and network hardware are considered. The research includes a review of graphical user interfaces and current operating systems. Actual system configuration experimentation is reported using various combinations of graphical user interface and application programs on the network. The conclusions presented are based on results of research and experimentation performed, with consideration given to network management issues.			
14. SUBJECT TERMS Local Area Network, Graphical User Interface (GUI), Operating Systems, Windows		15. NUMBER OF PAGES 72	16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18

Approved for public release; distribution is unlimited

GRAPHICAL USER INTERFACE NETWORK APPLICATIONS

M. J. Lanni
Lieutenant Commander, United States Navy
B.S., Monmouth College, 1978

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY MANAGEMENT

from the

NAVAL POSTGRADUATE SCHOOL
March 1995

Author:

M. J. Lanni

Approved by:

Norman F. Schneidewind, Advisor

William B. Short, Co-Advisor

David R. Whipple, Chairman
Department of Systems Management

ABSTRACT

This thesis presents a research study of configurations for installation of graphical user interface and software applications on a local area network. Requirements for a graphical user interface on a local area network are discussed, specifically for the Systems Management Department Laboratories at Naval Postgraduate School. Compatibility and interoperability of network operating system, PC operating system and network hardware are considered. The research includes a review of graphical user interfaces and current operating systems. Actual system configuration experimentation is reported using various combinations of graphical user interface and application programs on the network. The conclusions presented are based on results of research and experimentation performed, with consideration given to network management issues.

Accesion For	
NTIS	CRA&I <input checked="" type="checkbox"/>
DTIC	TAB <input type="checkbox"/>
Unannounced <input type="checkbox"/>	
Justification	
By	
Distribution /	
Availability Codes	
Dist	Avail and/or Special
A-1	

TABLE OF CONTENTS

I.	INTRODUCTION	1
	A. OBJECTIVES	1
	B. SCOPE AND METHODOLOGY	1
II.	BACKGROUND	3
	A. THE SYSTEMS MANAGEMENT LABORATORIES	3
	B. THE TOKEN RING LOCAL AREA NETWORK	5
	C. GRAPHICAL USER INTERFACES	7
III.	OPERATING SYSTEMS	9
	A. DESKTOP OPERATING SYSTEMS	9
	B. MS-DOS OPERATING SYSTEM	10
	C. OS/2 OPERATING SYSTEM	19
	D. WINDOWS NT OPERATING SYSTEM	26
IV.	GRAPHICAL USER INTERFACES	35
	A. ABOUT USER INTERFACES	35
	B. WINDOWS GUI	38
V.	GRAPHICAL USER INTERFACE CONFIGURATIONS	45
	A. CONFIGURING WINDOWS ON A LAN	45
	B. ALL WINDOWS FILES ON USER COMPUTER	46
	C. ALL WINDOWS FILES ON SERVER	48
	D. WINDOWS ON SERVER/USER FILES ON USER COMPUTER ..	50
	E. PERFORMANCE TEST RESULTS	54
	F. SUMMARY	56

VI. CONCLUSIONS AND RECOMMENDATIONS	59
A. PERFORMANCE IMPLICATIONS	59
B. CONFIGURATION RECOMMENDATION	60
LIST OF REFERENCES	61
INITIAL DISTRIBUTION LIST	63

I. INTRODUCTION

A. OBJECTIVES

The purpose of this research is to determine the optimal configuration for installation of graphical user interface and software applications on a local area network. The research will attempt to determine the requirements for graphical user interface on the local area network in the Systems Management Laboratories at Naval Postgraduate School. Compatibility and interoperability considerations will be discussed. Configurations will be considered which will allow maximum flexibility for installation of graphical user interface and applications on the local area network. Maximum flexibility refers to the ability to run a wide variety of user applications on the network in the graphical user interface environment.

B. SCOPE AND METHODOLOGY

This thesis includes discussion of requirements in the Systems Management Department Laboratories at Naval Postgraduate School. A review of current products available in the industry is included. A thorough explanation of user interfaces includes a detailed description of the Windows graphical user interface.

Research techniques include actual system configuration experimentation using the Systems Management Department computer laboratories in Ingersoll Hall at Naval Postgraduate School. Various possible configurations of graphical user interface will be tested on the network. Discussion will include performance results as well as administrative issues. Recommendations for configuration of the local area network will be made on the basis of test results, taking into account the issues of compatibility, interoperability, performance and administration of the network and its components.

II. BACKGROUND

A. THE SYSTEMS MANAGEMENT LABORATORIES

The Systems Management Department at Naval Postgraduate School (NPS) has established computer laboratories for use by students and professors. They were created as part of an effort to provide students "the necessary learning experience and the tools to reinforce the academic experience." (NPS, 1985) The original proposal for the laboratories states this objective would be accomplished as follows:

- ◆ Students will learn the fundamentals of local area networks and multi-user systems.
- ◆ Provide for enhanced learning through management simulation and gaming.
- ◆ Used as a test bed for student research in local area networks and distributed systems.
- ◆ Ready access by faculty and students to a library of on-line software packages.
- ◆ Allow increases in productivity by producing student theses. (NPS, 1985)

1. Requirements

The architecture of each laboratory was determined by its expected use. The laboratory in I-250 was set up originally for instruction in computer literacy, and initially had a broadband Ethernet PC Network installed. This was subsequently changed to an IBM Token Ring Network, identified as network 0TR. The laboratory in I-224 was set up to provide instruction in local area network technologies, and had an IBM Token Ring Network, a 3COM Baseband Ethernet Network, and a small Apple LocalTalk Network with PC to Mac communication. The Macintosh Network has since been moved to I-158, and I-224 now contains the Token Ring Network, identified as 4TR, and the 3COM Ethernet. The Token Ring networks in I-224 and I-250 are similar, and are directly

physically connected. Finally, the laboratory in I-158 was intended for research in Software Metrics, and also as a test bed for new applications and configurations, and has an IBM Token Ring Network and a Macintosh AppleTalk Network. The user computers on these LANs which have TCP/IP software installed are also connected to the Internet and to the campus backbone through the Cisco Router in the Computer Center.

Requirements for the laboratories are determined by the mission statement above as well as student and faculty needs. With the plethora of applications software available on the market today, we can anticipate multifarious software requirements for instruction and research. The optimal network configuration will include hardware and software which will allow for maximum diversity of user applications in the academic environment

2. Compatibility/Interoperability

Software programs and graphical user interfaces on the network must be easily usable by students and faculty in fulfillment of the stated missions of the laboratories. Some of the major issues which must be discussed and understood prior to changing the configuration or adding additional application programs are compatibility and interoperability of application programs with the graphical user interface shell, the workstation operating system and the network operating system.

Closely tied to the requirements for compatibility and interoperability are considerations of performance and ease of installation. In order to be fully utilized by the students and staff, the network must provide for new applications to be installed quickly and with minimum difficulty, and the applications must perform as expected. Thus performance standards, specifically response time and full functionality of applications programs on the network, must be tested and monitored.

As the networks are currently configured, most software is kept in the server and sent to the client or user computer upon request. A primary objective is to keep maintenance and installation simple. This means that software changes or modifications should occur only on the server. A careful evaluation of new software occurs before

installation on the Token Ring Network. It is important to avoid losing present capabilities by improper installation of additional software.

3. Flexibility

The combination of workstation operating system, network operating system, graphical user interface and application programs must allow sufficient flexibility for addition of application programs which may or may not operate under the graphical user interface shell, and may be written to work with one specific operating system/network operating system. The ideal configuration will provide the greatest flexibility for various combinations such that all software applications which may be required by students and professors in the future will be able to be integrated into the network.

B. THE TOKEN RING LOCAL AREA NETWORK

Token passing is a control technique which guarantees an upper limit on response time, i.e. deterministic performance. (Schneidewind, 1986) This technique is used in a ring topology in the Systems Management local area networks. It uses a control message called the token for granting permission to a node to transmit on the network. Generally, this method has better performance at high loads, because of no collisions. (Schneidewind, 1986)

1. The IBM 16-Mbs Token Ring

Since the focus of this research is on network applications for use by the students and faculty in an instructional setting or on an individual basis in the computer laboratories, further discussion will focus on the Token Ring Networks 4TR and 0TR. These networks are the most widely used and have the greatest need for expanded graphical user interface in the use of applications software.

The computer laboratory networks are IBM Token Ring Networks. The IBM system is a powerful LAN hardware platform capable of handling hundreds of workstations without loss of capability. The IBM Token Ring Network uses a star-wired ring topology and follows the baseband signaling and token passing protocols of the IEEE 802.5 standard. (Schatt, 1992) The network uses shielded twisted pair, designated by IBM as Type 2 cabling. The Token Ring Network operates at 16 megabits per second (Mbs) and supports up to 260 devices.

The 16 Mbs network uses a technique known as "early token release." A workstation may transmit a token immediately after sending a frame of data, instead of waiting for the return of the transmitted packet. The use of more than one token on the same ring at the same time increases network efficiency and speed.

2. Token Ring Hardware

The Token Ring Network requires an adapter card in each user workstation. The adapter card exchanges data buffers and control blocks with the workstation RAM to provide for operation of the network. The adapter card in each workstation handles token recognition and data transmission.

The multistation access unit (MAU) is a wiring concentrator which permits up to eight network workstations to be inserted on the ring. Each MAU contains ten connector jacks. Eight of these ports are used to connect network workstations, while the remaining two are used to connect other multistation access units.

3. Network Operating System

The Systems Management Department runs the PC LAN Program on the Token Ring Network. This hardware platform can also run LAN Server or Novell's NetWare Network Operating System. Token Ring Network uses a NETBIOS software program that is loaded into each machine. (Schatt, 1992)

C. GRAPHICAL USER INTERFACES

Graphical user interface is a feature which enables users to view applications in various "windows" on their screens. (Schatt, 1992) Graphical user interface, also known as GUI, may be provided through an operating system or through a shell which runs over an operating system, such as Windows over DOS. GUI are attractive to users because they provide intuitive interface with the system or application in use. Another appealing feature of GUI is that they can switch from one active program to another, giving the appearance of multi-tasking. Currently there are several applications on Network 0TR that run under Windows. With the integrated network concept, the GUI program is required to be installed on only one server. However, this concept is subject to further testing to evaluate the penalty of Windows on server versus user computer. This research will attempt to explore the different configurations for installation of Windows and various other options available for adding expanded GUI capability to the Token Ring Network.

III. OPERATING SYSTEMS

A. DESKTOP OPERATING SYSTEMS

The operating system program manages all computer system resources and operations. It provides the user interface to the computer and an interface for user programs to the hardware. The responsibilities of the operating system include managing the memory installed in the computer, the hard disks and file systems, and the processor.

An operating system can be distinguished by the number of programs it can run concurrently. An operating system that can run a single program at a time is known as a single-tasking system, while one that can execute multiple programs simultaneously is known as multitasking operating system.

Multitasking can be supported by an operating system in several ways. The difference between types of multitasking is in the way applications pass control from one to another. The following list describes the three main types of multitasking:

- ◆ *Timeslice Multitasking*. In this method of multitasking, resources are moved between programs on a fixed schedule. The operating system relies on timers built in to the processor hardware to stop a running program and reactivate the operating system. Even if an active program does not need resources, it receives them. This method can waste processing resources.
- ◆ *Cooperative Multitasking*. Each program can request resources, but other applications are not forced to respond to these requests. Cooperative multitasking is used to perform background operations with programs designed to work in this environment. Cooperative multitasking works by giving control to a program to perform one task or event and relying on that program to turn over control to the next application when the task is accomplished or when a maximum time limit has passed. This method relies on programmers developing applications that are well behaved in this environment.
- ◆ *Preemptive Multitasking*. Each program is assigned a priority and can compete for use of processing resources. This capability enables the operating system to evaluate whether the process that was stopped should be allowed to

continue or another process should be activated. This type of multitasking is generally available only with sophisticated operating systems.

The concept of multitasking involves multiple processes running at the same time. Every process is given its own memory space to run, and is managed as an independent entity. A thread is similar to a process, but shares memory space with the parent process. Each executable path through a process is a thread. A process can run threads in parallel for faster execution.

File systems perform file and disk management in the operating system. These utilities are closely related to input/output control, which involves movement of data between peripheral devices, the processor, programs, and RAM (Long, 1992). Handling input and output to a specific device is handled by a "device driver." This may be hardware or software, and is usually provided by the device manufacturer or software vendor (Tannenbaum, 1992). The device driver interfaces the hardware or software to a particular operating system.

Many operating systems include support for a graphical user interface (GUI). This is graphics-oriented software that controls the user interaction with the computer and the operating system. Some GUIs, such as the Finder in the Macintosh operating system, is an integral part of the operating system structure. Other GUIs, such as Windows, is a graphical environment built on top of an existing operating system (DOS). Graphical user interfaces will be explored in the next chapter.

B. MS-DOS OPERATING SYSTEM

DOS workstations make up the majority of computers running on local area networks. DOS is a single-tasking, single user operating system sold originally by Microsoft Corporation. It is by far the best-known operating system for personal computers.

MS-DOS is both the simplest and most inexpensive operating system. It contains no graphics, no networks and virtually no applications. Thousands of applications and add-on programs have been designed to improve basic MS-DOS functionality and usability.

MS-DOS was introduced in 1981 by IBM and Microsoft Corporation as an operating system for the new IBM PC, which operated on the Intel 8088 microprocessor. IBM also sold a version of the operating system labeled PC DOS. Both operating systems were developed for use by a single user in a single tasking environment. The original MS-DOS operating system consisted of 4,000 lines of assembly language code and occupied 12K of RAM (Tannenbaum, 1992).

As Intel introduced newer processors, MS-DOS continued to support the original Intel 8088, and the associated basic input-output system (BIOS) created by IBM. Today, MS-DOS can still run on an 8088 processor and on the 8086, 80286, 80386, 80486 and pentium. (Day, et al, 1993)

Microsoft, IBM and Novell have recently been engaged in lively competition for the DOS market share. Several DOS upgrades were released in 1994 as the software industry's three main DOS providers engaged in a "game of DOS one-upmanship." (Bertolucci, 1994)

MS-DOS 6.21 is the most recent version. IBM's PC DOS 6.3 is a comparable operating system based on the same programming code. The standard DOS commands and programming functions are identical in both operating systems, but IBM's optimized version has slightly faster character input/output and batch file processing. The major difference between MS-DOS and PC DOS is the utilities that are offered. PC DOS 6.3 offers integrated data compression as one of its key features, which is not included with MS-DOS 6.21 (Bertolucci, 1994).

The newest DOS on the market is Novell DOS 7, which emulates MS-DOS while offering built-in networking via Novell's NetWare peer-to-peer network operating system (Mendelson, 1994). Novell DOS is built upon Digital Research DOS (DR DOS).

Although Novell DOS is highly compatible with its Microsoft and IBM counterparts and uses almost identical commands, its program code was newly written by Novell. The operating system offers a new memory management feature, a full implementation of disk compression software, and password protection for files, directories and the entire system (Mendelson, 1994).

Even with the advent of new flavors of DOS, Microsoft's MS-DOS remains the preeminent PC operating system in the world. In addition to the relatively low cost and the fact that it comes bundled with almost every PC sold today, the platforms that can support MS-DOS are also inexpensive. Especially with the extensions offered by Windows 3.1, MS-DOS continues to reign as the leader of PC workstation operating systems. The three leading environments, native DOS, DOS Shell and Windows 3.1 can provide most users with a sufficient level of functionality and service.

1. Process Management

The original design of MS-DOS included a fundamental assumption that a personal computer is used by one person, who is unlikely to need multiple tasks running simultaneously. Thus, a single-tasking system seemed adequate. Today, MS-DOS can meet the needs of more sophisticated users with the addition of certain application software to give the appearance of being a multitasking system.

A key reason why MS-DOS is a single-tasking operating system is the non-reentrancy of DOS functions. *Non-reentrancy* means that when a program uses an operating system service, the service must run to completion before it is started again (Day, et al, 1993). Non-reentrant code must be executed from start to finish before a particular module of code may be called and executed again.

Processes in MS-DOS are initiated by one of three types of executable files. These are command files, designated by the file extension ".COM"; execute files, designated ".EXE"; and batch files, designated ".BAT." (Day, et al, 1993). The ".BAT" files are composed of listings of MS-DOS commands. The "COMMAND.COM" file contains

DOS internal commands, and loads other operating system and applications programs into RAM as they are needed or directed by the user (Long, 1992). External commands are transient programs stored as files on disk, and brought in as needed.

DOS processes may also overlay or terminate-and-stay-resident (TSR). A running process can load a .COM or .EXE file called an *overlay* into RAM and then execute the file. Upon completion control is returned to the running process. In *TSR* a program stays in memory without being overwritten.

Although MS-DOS is not a multitasking system, the addition of software applications can give the appearance of multitasking by moving very quickly between processes. Microsoft Windows installed on a DOS operating system gives a good illusion of multitasking on a single-tasking operating system.

2. File Systems

The MS-DOS file system is organized hierarchically. Each file system starts at a single root location and splits into branches called directories. The directories have individual files, which are analogous to the leaves on a tree (Day, et al, 1993).

MS-DOS uses a simple naming convention in which each file has an eight character name and a three character extension. The file-name extension generally indicates the purpose of the file, although only the three previously mentioned, .COM, .BAT and .EXE, have any significance to the operating system. Other file extensions may be .DAT for data files, .DOC for document files, .TXT for text files or .OBJ for object files.

A computer hard disk is divided into partitions of information with a drive letter representing each partition. The partitions start with a boot sector, which gives MS-DOS information about the start-up process and a File Allocation Table (FAT). The FAT contains information about storage space on the disk. Every FAT entry is a 16-bit data word that indicates whether the block is free, bad, end of file, or points to the next block of the file. The FAT is the basis of the DOS file system.

An MS-DOS directory entry stores the initial block of each file. The directory entry is 32 bytes long, and consist of the file name, extension, time, date, first disk block and file size. Attributes are in the middle of the extension field. Attributes offer file protection for one user on the system.. The following is a list of the bits that may be found within the attribute byte:

A	<i>Archive</i>	indicates a file was modified
R	<i>Read-Only</i>	file cannot be modified
S	<i>System File</i>	prevents deletion
H	<i>Hidden</i>	not shown in a directory
V	<i>Volume</i>	volume label
D	<i>Directory</i>	directory, not a file

Because MS-DOS is a single-user system, files cannot be protected from unauthorized access without additional software.

Device drivers are part of the file system of MS-DOS. Device drivers provide basic input and output (I/O) for DOS. The device drivers are hardware specific programs that interface with DOS. They are located in the CONFIG.SYS file. To add a new device the user can simply add a command line to this system file. The device driver command is *DEVICE = name of device driver* (Microsoft, 1992).

The best-known limitation of DOS is that it directly supports only 1M of memory. The lower 640K is known as *conventional memory* and is used for application programs and the DOS kernel. It is also used for the network operating system (NOS) kernel if it is active. Memory managers are used to load drivers into the *upper memory* (640K to 1M), allowing more space in conventional memory. Additionally, extended memory and expanded memory can overcome the 1M limitation. *Expanded memory*, located above the 1M limit of conventional memory, is a mechanism which allocates a 64K page of unused memory in the upper 384K of conventional memory. This area is used to swap pages in

and out of memory above the 1M limit. *Extended memory* uses memory above 1M through use of a memory manager. Advantages are that programs too large to fit in the DOS 640K space can be run, and valuable conventional memory space can be conserved by loading TSRs into high memory.

3. Native DOS

The COMMAND.COM file provides the user interface to DOS. This file processes all entries made from the keyboard at the DOS prompt. This file has a number of commands stored internally, and it will search for files that match the spelling of commands typed by the user. Native DOS is not a graphical user interface, and the DOS command prompt is the familiar C:. Nearly 70 commands can be entered from the DOS prompt. The commands can help the user maneuver through the file system, view the contents of a file directory or a file, change the attributes of files, check or set the computer's date and time, format disks, backup disks, copy, move or delete files and directories or run an application program.

4. DOS Shell User Interface

The DOS Shell interface is an alternative to the command prompt interface. It is a screen oriented interface composed of a series of menus which allow the user to perform many basic functions using a mouse instead of a keyboard for program input. The DOS Shell provides the ability to switch quickly between different DOS programs. When more than one program is started with the DOS Shell, one program is automatically suspended when the second or third is activated. However, this task-switching capability does not offer any type of background operation. A suspended task cannot be doing anything. Therefore any program that must be kept running may not be able to work within the DOS Shell task switcher.

The DOS Shell is divided into five areas:

- ◆ *Title Bar.* The Title Bar shows the user that they are in the MS-DOS Shell with the title "MS-DOS Prompt, MS-DOS Shell."
- ◆ *Menu Bar.* The Menu Bar lists a number of pull-down menus which can be accessed to perform certain tasks. The menus include "File," "Options," "View," "Tree," and "Help."
- ◆ *Directory Tree.* The Directory Tree area shows a graphical representation of the current directory structure.
- ◆ *File List.* The File List Area shows a list of files in the current directory.
- ◆ *Program List.* The Program List area contains a predefined list of programs that can be executed from the DOS Shell. This area can be modified by the user as additional programs are added to the computer.

The DOS Shell is a very limited single tasking graphical interface. True graphical user interface with DOS is provided by Microsoft Windows. Windows is actually an operating system environment program for DOS. This is the most common enhancement to the DOS interface today.

5. The Three DOSs

A brief review of the three DOSs on the market today should begin with MS-DOS, the standard. Version 6.2 was released in early 1994 (Roberts, 1994) as an upgrade to MS-DOS 6.0.

The major improvements introduced with the Version 6.0 were memory management, configuration, and virus protection. The program placement feature made it possible to load programs into specific locations in upper memory. Another useful new feature was *MultiConfig*, a fairly simple program built into the operating system that lets a user enter multiple configurations into a single AUTOEXEC.BAT/CONFIG.SYS, and allows a specific setup to be chosen with two keystrokes (Minasi, 1993). The introduction of *Interlnk* allowed for the connection of two PCs parallel port to parallel port with a device driver being loaded. In this way one of the machines acts like a LAN

server. The primary use for Interlnk is for downloading a laptop client to a home server PC (Minasi, 1993). For virus protection, Version 6.0 offered *MSAV*, a virus scanner and *Vsafe*, a virus shield, both of which were based on the very popular Central Point Antivirus. *MSAV* reports and flags viruses by looking for distinctive strings in program files or by creating checksums for each program file which can then be recomputed and compared later to previously created checksums. *Vsafe* is a program that runs continuously in the system, trying to keep data safe by monitoring efforts by programs to go resident in memory (become TSRs). *Vsafe* also monitors attempts to use the BIOS's FORMAT command, normally only used by the DOS FORMAT command, and will alert the user to the request for format operations before BIOS can carry them out, offering a chance to override the request (Minasi, 1993). *DoubleSpace*, the new disk compression routine, had a problem when initially introduced with Version 6.0. It was writing data to fixed disk areas prior to checking if the area was good or bad. This was the major problem with 6.0, which led to the hasty release of MS-DOS 6.2.

The latest version of Microsoft's best-selling operating system includes several new utilities in addition to a "fix" for *DoubleSpace*. *DoubleGuard* verifies data in memory before writing to disk. It is a checksum-based system that identifies corruption in *DoubleSpace* memory buffers before data is written to disk. If this verification fails, the system is shut down to limit further data corruption (Roberts 1994). A new diagnostic routine in the HIMEM.SYS file includes the ability to single-step through AUTOEXEC.BAT and bypass *DoubleSpace* at startup (Wolverton, 1994). Version 6.2 comes equipped with *ScanDisk*, a disk diagnostic and repair utility which can repair cross-linked files, test the integrity of each sector and repair or lock out bad sectors. The *SMARTDrive* disk cache utility provides the ability to cache CD-ROM drives, and will flush the cache to disk before returning the user to the C prompt. Other programs enhanced in MS-DOS 6.2 include *Defrag*, which is able to defragment disks that hold as many as 20,000 files, and *Windows Undelete*, which will no longer allow users to enter invalid filenames.

IBM's PC DOS is very similar to MS-DOS with some support for pen computing and PCMCIA cards. PC DOS 6.1 was completed before a shared technology agreement between IBM and Microsoft expired in late September 1993, so most of the base code is identical (Wolverton, 1994). Standard DOS commands and programming functions are the same as those in MS-DOS. It looks and feels like MS-DOS, just as previous releases of both operating systems have been nearly identical. The additional utilities include the *PenDOS* program, for systems with a stylus and tablet. This feature lets the user use a stylus as a mouse replacement, for writing DOS commands or for entering data into applications. The PCMCIA functions offer DOS and Windows support for fax, modem, memory and hard disk cards. The *SuperStor* compression is IBM's version of DoubleSpace disk compression. (Mendelson, 1993)

Novell DOS 7.0 emulates MS-DOS and offers built-in networking via the Novell NetWare network operating system. The program offers a full implementation of Stacker Version 3.1, a disk compression software. A multitasking utility lets the user download files or sort a database in the background while using a word processor in the foreground. And Novell's DOS introduces a new memory management feature named DOS Protected Mode Services (DPMS). It allows drivers and TSR programs to load much of their code above the first megabyte of memory, but it does not use 64K of memory for a page frame (Mendelson, 1994). DPMS lets the network software, disk cache, CD-ROM extensions and Stacker disk compression reduce their footprint in the first 1M of RAM. It also works with third party memory managers such as QEMM. Novell's DOS is more technically sophisticated than Microsoft's and IBM's and is particularly attractive to users of peer-to-peer LANs.

6. Summary

DOS is best suited to a single user with a single system performing a single task at a time, in the environment which DOS was originally created to support. MS-DOS is no doubt the preeminent PC operating system in the world. With extensions offered by

Windows, MS-DOS will continue to lead the field of PC workstation operating systems. Combined with various environments, an MS-DOS based system will meet virtually all computing needs of the single user.

C. OS/2 OPERATING SYSTEM

The IBM OS/2 operating system is a true 32-bit operating system, using the powerful multitasking capabilities of the Intel processors. It is a single user, multitasking operating system. It can run most MS-DOS programs and Windows programs. Its strength lies in its capability to support multiple processes and multiple threads per process.

Development of OS/2 began in 1984 when IBM introduced the IBM PC-AT. The first version was introduced in 1987, and had no graphical user interface and no mouse support (Day, et al, 1993). OS/2 was designed to replace DOS by breaking the 640K barrier and by providing true multitasking with graphical user interface. Today's OS/2 promises "a better Windows than Windows, and better DOS than DOS." (Day, et al, 1993)

OS/2 is a true timeslice multitasking operating system. It also offers cooperative multitasking and preemptive multitasking. The capability to multitask applications extends not only to OS/2 applications, but also Windows applications and MS-DOS applications.

OS/2 tries to be all things to all people. In addition to the standard interface, it offers the Workplace Shell, an OS/2 command-line interface, an MS-DOS command-line interface and a Windows environment. IBM has done remarkably well with application compatibility, considering the differences in these operating systems.

The current versions are OS/2 Warp Version 2.11 and Version 3.0. OS/2 Warp Version 2.11 is for installation on a machine without Windows and offers Windows capability. OS/2 Warp Version 3.0 is for workstations that already have Windows or Windows application compatibility is not desired (Phaniraj, 1994). They are advertised to

run 100% faster than their predecessor, OS/2 2.1 (Abernathy, 1994). The operating system is based on an object-oriented architecture. This system-wide object orientation has helped IBM improve OS/2's versatility, offering dozens of environment and operating system attributes that can be modified for specific performance.

1. Process Management

A *session* in OS/2 is any mode that is running on the computer. The three modes of operation are the OS/2 mode, the DOS mode and the WIN-OS/2 mode (for Windows). All OS/2 programs running directly from the OS/2 Workplace Shell are in a single session. Windows programs which are running are in another session, and any DOS programs running are in yet another session.

OS/2 views programs running on the computer as a collection of threads with different priorities. Programs may have one or more than one thread of execution. For a particular program, each thread is treated with equal priority. OS/2 recognizes three classes of priorities (Day, et al, 1993):

- ◆ *Interactive*. High or urgent priority. Programs interacting with the user react immediately to input. Interactive priority gives the user immediate response.
- ◆ *Foreground*. Normal priority. Programs active in the current session use interactive and foreground priority. Foreground priority is always applied to the visible screen where the user is working. Foreground can be run as the second-highest priority to interactive.
- ◆ *Background*. Low priority. Any session that does not contain the program with the interactive priority is placed in the background and gets the lowest priority. Background priority is assigned to the non-visible screens and receives CPU time only when the interactive and foreground priority programs are not busy.

OS/2 will give I/O-bound processes a higher priority than CPU-bound processes to ensure good device utilization. OS/2 assigns each process a Process Identification

Number (PID) to keep track of a process and its threads (Dietel, 1990). Each process has its own address space. The operating system ensures the address spaces do not conflict between processes.

A process can change priority within a session depending on what it is doing and how long it has been waiting for CPU time. A process running in a session that does not have interactive priority always receives lower priority than a session with interactive priority. If more than one thread of execution has the same priority, each is given an equal share of processor time.

The DOS mode and WIN-OS/2 mode are possible because they run as a subset of the Application Program Interface (API) of OS/2 called the *family API* (Dietel, 1990). Application programs written using the family API can run on DOS or OS/2 operating systems.

The operating system supports all three types of multitasking in the OS/2 and WIN-OS/2 modes. In *timeslice multitasking*, each process or program running is given an equal share of CPU time. The more programs that are running the less time is available for a single program. OS/2 *preemptive multitasking* is based on a predetermined priority scheme. OS/2 supports *cooperative multitasking* for Windows applications. In cooperative multitasking, the running program returns control to the operating system when it seems logical to do so, even if the process still has more time to run in its timeslice. (Day, et al, 1993) In this way, programs that are not busy do not take much processor time.

DOS applications do not use cooperative multitasking. OS/2 does not support Windows 386 Enhanced mode for DOS because it users hardware-specific features (Day, et al, 1993).

OS/2 supports dynamic linking of processes located in *dynamic link libraries (DLLs)* (Dietel, 1990). Dynamic linking allows a program to call a subroutine that has not been linked to executable machine code in memory. In this way new subroutines can easily be incorporated into existing programs. DLLs also contain the OS/2 application

programming interface (API). This allows software developers to write DLLs without modifying the operating system and supports an open system architecture.

OS/2 allows the user wide scope in process management. The status of a process can be displayed to the user with the "PSTAT" command (Dietel, 1990). Some status information available from this command includes DLLs, PID and shared memory.

2. File Systems

OS/2 has two types of file systems which offer significant advantage over MS-DOS. These are the DOS-compatible File Allocation Table (FAT) and the High Performance File System (HPFS).

The FAT file system was designed for small disk sizes, therefore the use of a FAT file system becomes very slow as the drive size increases. The FAT contains information on every block of storage on the disk. The FAT continues to be supported because of DOS program compatibility.

The High Performance File System (HPFS) supplies increased performance and functionality, but no compatibility exists between the FAT and HPFS file systems, and DOS cannot read an HPFS partition on a hard disk. The HPFS is a hierarchically organized file system that can be represented graphically in a tree-like organization. The file system starts at the root directory and then breaks into files and directories. The naming system is flexible, with file names which can be as long as 254 characters. There is no restriction on the location of periods in the file name, and it is case sensitive (Day, et al, 1993). Like DOS, OS/2 uses file endings to indicate batch and system files that are executable. EXE, COM and SYS indicate executable files, and CMD indicates batch files.

In HPFS the disk is divided into 8-megabyte bands. *Free-space bit maps* are located at the end of odd-numbered bands and the beginning of even-numbered bands. Every sector on an HPFS disk is marked as used or full in the free-space bit map. This arrangement enables large files to have sections up to 16 megabytes. Each file and directory on a disk is identified with an *Fnode*, the basic allocation pointer used in HPFS.

The Fnode uses a 512 byte block to store internal file system information, such as extended file attributes, access control lists, file length, the first 15 characters of the file name or directory, and the allocation structure that defines where the file or directory is on the disk (Day, et al, 1993). To save time accessing files, the Fnode that describes a file is physically located near the file on the disk. In this way, after the location information is read, the disk heads do not have far to travel to read the file.

In the Fnode, a file is defined as a collection of one or more strips of data on the disk. Each strip consists of a starting point and a length. A single Fnode can hold information for eight strips of data, each strip up to 16 megabytes long. If the file has more than eight pieces, the Fnode uses a process called *indirection* to hold pointers to other structures that contain the location and lengths of the data strips.

The HPFS uses a binary tree structure to lay out various parts of the directory structure. Like the DOS file system, HPFS starts at a single root location. The *SuperBlock* specifies the location of the root Fnode at the beginning of the disk. The SuperBlock is located just after the boot block so the location of the root Fnode can always be located. OS/2 begins looking in the root directory Fnode to find the root directory file when a program requests access to a file. After the root directory file is located, OS/2 opens it to look for the file specified by the program.

All directory files are 2K long and can store a variable number of directory entries depending on the length of the names in the directory and the number of additional attributes stored in the directory entry. After the root directory block becomes full, HPFS creates another layer of directory blocks and the root block becomes a list of directory blocks. HPFS sorts directory entries according to their names, rather than random assignment to the blocks. With this system, OS/2 can locate files, even on large disks, with very few reads.

Disk read caching is the most common type of disk caching, and can be used with the DOS FAT file system as well as HPFS. This consists of the file system reading 2,048 byte chunks of requested files into RAM. The requested part of the file is passed to the

requesting program, and the remainder is stored in the RAM disk cache (Day, et al, 1993). If the program requests more of the file, the needed information is likely to have been read from the disk already and to be in the RAM cache. HPFS does one step more than DOS in disk read caching by enabling large sections of the disk to be stored in memory and placing information in the file directory entry to indicate typical use patterns for the particular file.

Disk write caching involves a program writing data to a RAM write cache, which is written to the disk after the write cache gets full. This is known as *lazy write*. This function can be controlled by the user with the "CACHE" command in the CONFIG.SYS file. Write caching must be used carefully, because while the data is still in the RAM cache, it is not permanent data. The HPFS file system uses a *SpareBlock* to backup write caching for error recovery and system failures such as power loss (Day, et al, 1993). The SpareBlock is located just after the boot block and the SuperBlock. If a write failure occurs, HPFS writes the data to an emergency area whose location is kept in the SpareBlock. The OS/2 Workplace Shell GUI will notify the user when a write failure has occurred.

Every read of the disk is preceded by consultation of an error table, called a *hotfix map*, to check whether the part of disk being read is good or bad. To keep track of the integrity of the disk structure, a binary flag is set whenever any file is open on the entire volume. The flag is reset only when all files are closed, for instance with normal shutdown of the system. The flag is checked upon boot-up, and if the flag is still set the system will run CHKDSK program to attempt to reconstruct the disk before resuming system usage. (Day, et al, 1993).

3. Environment

OS/2 offers a variety of environments to the user. These include the Workplace Shell, an OS/2 command-line interface, an MS-DOS command-line interface and a

Windows environment. The theory is that a user can now switch to OS/2 and protect their investment in DOS and Windows programs.

The Workplace Shell is the default interface for OS/2. This environment allows the user to run multiple sessions of OS/2, DOS and Windows simultaneously. Each type of program is maintained in a separate folder for easy access. A session can be started by clicking on the icon.

The Workplace Shell identifies every file by its use and represents them as objects. Four types of objects appear on the Workplace Screen:

- ◆ *Program Objects.* These are icons that represent programs on the system, such as spreadsheets or databases.
- ◆ *Folder Objects.* These are collections of other objects. Folders may hold other folders to provide a user-defined hierarchy. The primary folder is the System Folder that forms the top of the hierarchy.
- ◆ *Data Objects.* These are files created and used by the programs on the system. Data objects include text, spreadsheets, sound and video.
- ◆ *Device Objects.* These represent the physical peripheral devices on the computer. Examples are printers, plotters and modems.

The Workplace Shell is consistent with IBM's *System Application Architecture (SAA)* plan which includes a common programming interface, common communications support and common user access (Day, et al, 1993). The primary goal of SAA is to present a common interface across many systems. In this way the user should face fewer migration issues when moving to a new or larger system as requirements increase.

An alternative to the Workplace Shell is the Windows environment. OS/2 has Windows embedded in the operating system (Day, et al, 1993). Programs running in Windows must be run in the standard mode, because OS/2 does not support Windows enhanced mode. Windows programs can be run as full scale Windows applications or as part of a WIN-OS/2 session, in which several Windows applications may be active at one time.

The OS/2 command prompt and the DOS command prompt interfaces are also offered by the operating system. Most of the commands used in OS/2 are the same as the ones used in MS-DOS. The OS/2 command line interface supports the Restructured Extended Executor (REXX) batch programming language. The REXX programming language is SAA compliant (Day, et al, 1993).

4. Summary

The design of OS/2 under IBM's System Application Architecture (SAA) supports the common set of application programming interfaces and compilers across many different hardware platforms. This ensures compatibility with some of IBM's other products.

There are some disadvantages to OS/2. Its size and complexity have caused consumers to move slowly to OS/2. The competition of Windows, Windows NT, Macintosh System 7, and UNIX provides consumers with many operating system choices and makes a large-scale move to OS/2 difficult to justify.

Advantages include compatibility and scalability, being equally operable on the most humble word processor as on the notebook PC and the high-end desktop system (Abernathy, 1994). It has met the challenge of supporting applications that currently are popular and that run under DOS and Windows. It also provides a powerful base operating system for developers to use in future application development.

OS/2 is not the only 32-bit operating system being marketed. In addition to UNIX products, Microsoft has released Windows NT which will be discussed in the next section.

D. WINDOWS NT OPERATING SYSTEM

Windows NT (new technology) is an entirely new operating system built to surpass the limitations of the DOS operating system. It was designed as a state-of-the-art operating system to satisfy the computing needs of large organizations, but to be

compatible with the installed base of DOS and Windows applications. The result is a 32 bit operating system that supports 16 bit DOS and Windows applications (Feldman, 1993).

Windows NT includes built-in networking capabilities, a C-2 security rating and the ability to run on many platforms. It can use different alphabets in support of international computing (Myers, et al, 1993). True multitasking ability and enhanced graphics within the Windows interface are the major features of the Windows NT operating system.

In addition to DOS, Windows and Windows NT applications, NT also runs a limited set of OS/2 applications. It is also capable of running POSIX 1.0 compatible applications. The level of support provided is small, and for the most part represents a gesture of providing cross-platform compatibility.

One of the most important features of Windows NT is its ability to run on non-Intel processors. In addition to Intel's family of 386, 486 and Pentium processors, NT supports the high-performance RISC (Reduced Instruction Set Computer) processors.

Windows NT requires at least a 25 MHz 386 processor and 12 MB of RAM, although this is an absolute minimum and is really impractical. It also needs 80 MB of disk space for the complete operating system, including networking, OS/2 and POSIX support. For many users the huge disk space requirements are taxing on their systems. The complete Software Development Kit requires additional 40 MB, and swap files for virtual memory needs another 10-20 MB. For the user who will be developing or simply running applications 100-140 MB of hard disk space will be required just to get started. Disk compression cannot be used because the operating system is totally incompatible with the Stacker and DoubleSpace compression utilities.

The latest release is Windows NT Version 3.5, released in September 1994. This release includes two products, Windows NT Server 3.5 for LAN servers, and the standalone version for technical and financial workstations (Markoff, 1994).

1. Process Management

The operating system operates a sophisticated preemptive multitasking system and a multithreading capability which enables NT to do more things at once. The multithreading enables many programs or processes to be broken down into two or more threads, each of which can be executed concurrently. NT also supports multiprocessing, where two or more processors can work together to execute applications. By combining multiprocessing and multithreading, each processor could run different portions of the same application, or entirely different applications in order to gain performance improvements. However, most applications must be written specifically with multiprocessing in mind. Virtually no existing Windows applications can take full advantage of multiprocessing without extensive redesign (Feldman, 1993).

Windows NT runs most existing DOS and Windows applications without modification by using emulators. To run DOS applications, the emulator, called the *Virtual DOS Machine (VDM)*, completely reproduces the PC hardware architecture in software as well as the DOS operating system. Thus DOS applications run "inside" VDM under Windows NT. Windows applications also run on NT by using an emulator called *Windows on Win32 (WOW)*. This emulator runs on top of the VDM and supports most standard and enhanced mode Windows applications. Multiple applications can run simultaneously inside WOW, but since the WOW environment so closely duplicates Windows, an application crash within WOW will likely crash other running applications. Unlike DOS/Windows-based systems, however, a crash within WOW will not affect other non-Windows applications. (Feldman, 1993)

Windows NT has two running modes for the processor. In the *kernel mode*, the system allows access to hardware and system memory. Only the system itself uses this mode. Applications always run in the *user mode* with fewer privileges. Applications utilize system resources by calling system routines. The system then switches to kernel mode, executes its own service routine, switches back to user mode, and returns control

to the application. Collectively, the kernel mode portions of Windows NT are called the *NT Executive*. (Myers, et al, 1993)

To support different *application program interfaces (APIs)*, Windows NT relies on a set of user mode subsystems. NT includes four subsystems to support Win32, POSIX, MS-DOS and 16 bit Windows (WOW). The environment subsystems that support different APIs are servers, and programs that use a particular API are clients. The API servers are "protected" subsystems, because each runs as a separate process with a private address space. In addition to the protected environment subsystems, a security subsystem logs users on and keeps track of their accounts.

2. Object Management

A Windows NT object is a data structure with two main parts, a header and a body. Objects of all types have similar headers, but the objects themselves come in different types depending on what is represented. All objects are maintained by the object manager.

Most parts of the system create object types to suit their needs. Executive objects are available in user mode. Some more common executive objects are:

- ◆ *Process*: A running program.
- ◆ *Thread*: An executing unit within a process.
- ◆ *Section*: An area of memory shared between processes.
- ◆ *File*: Disk file or an I/O device open for access.
- ◆ *Port*: Place to deliver messages for a process.
- ◆ *Access token*: Contains security information that sets privileges for logged-on user.
- ◆ *Semaphore*: Counter controlling the number of threads that use an object at one time.

In order to distinguish between objects and keep track of them, the system names objects and stores the names in a list. The list of named objects resides in global memory so a process can locate and open objects made by another process. The object list is structured in a hierarchical tree of directories. Each supported file system, NTFS, FAT, HPFS and POSIX, creates its own hierarchy with rules understood by a driver in the I/O system. The self contained set of objects constitutes an object domain. The I/O system manages the objects in its own domains, creating one for each disk device. These device objects appear in the global object hierarchy, so the object manager can use the I/O system to read object names from the domain of a disk device object.

Windows NT uses a handle mechanism to avoid searching through the entire tree every time a program refers to any object. Each process stores pointers to all the objects it uses in a private object table. The process receives a handle indexing entries in its object table. The handle remains effective until the process closes the object.

In addition to acting as a pointer to an object, each entry in an object table holds information for object inheritance, access privileges and resource accounting.

3. File Management

The Windows NT file system is called NTFS (NT File System). It works with the Security Subsystem to keep track of security permissions for every file, subdirectory, and user. The NTFS also supports disk mirroring, in which two disk drives contain identical information, so that the mirror drive can take over in the event of a failure of the primary drive. NTFS improves disk performance using a technique called *disk striping*, in which files are written or *striped* across two or more physical drives.

Each drive can read or write a different block of data from the same file simultaneously. This greatly enhances disk function performance. With the use of parity, small data errors can often be corrected and data damaged by more severe hardware or software failures can be reconstructed.

In addition to NTFS, Windows NT is also compatible with the FAT file system and OS/2's HPFS.

4. Memory Management

Windows NT supports a linear memory address space of four gigabytes, two gigabytes for programs and two gigabytes for the operating system (Feldman, 1993). This method eliminates the use of various memory managers used with the DOS and Windows combination, such as HIMEM or QEMM. It also has a virtual memory manager which gives each program more room to run by combining available RAM with hard disk space. In this way it protects the system by keeping programs and system tasks from interfering with each other.

Windows NT is designed for newer CPUs that support logical addresses in the hardware. The significance of this is the user cannot circumvent the virtual memory manager and use a physical address to access another program's memory. No other program can interfere with the physical memory assigned to a program. The 2GB of logical memory for every application is a private address space. Thus, even if two processes happen to use the same logical address, neither will interfere with the other because each address is mapped to a different part of physical memory. One consequence of this feature is that programs cannot exchange addresses with each other for purposes of data sharing. Other system-supported mechanisms accomplish this.

When physical memory is full, the virtual memory manager begins paging sections of memory into a large hard disk file. For efficiency the memory manager moves blocks of 4K at a time. Advanced CPUs support pages as part of their built-in virtual memory capabilities.

Processes can share memory by creating *section objects*. A section object is an area of physical memory that has been mapped into two virtual address spaces (Myers, et al, 1993). If a section is very large, a process may open a small window in the section

called a *view*. The view can be moved along the full length of the section so that the process can read all the shared data.

Windows NT protects memory in several different ways. First, the distinction between kernel mode and user mode prevents non-system code from reading system memory. Second, the virtual address space firmly isolates programs from each other. Third, individual memory pages can be marked to limit access. And finally, section objects receive the same protection that all objects do, that no process can open an object handle without authorization.

5. Subsystems

The Win32 subsystem provides the user interface with Windows NT. The subsystem also coordinates the interactive operations of the other protected subsystems so that applications written for DOS, POSIX, OS/2, Win16 and Win32 can all share the screen.

The Win32 subsystem is an independent server process looping continuously and waiting for requests from client programs. The subsystem manages the screen for all subsystems. For character-based programs it creates a console window like the command shell. It tracks which window has the focus and which subsystem should receive input from the mouse and the keyboard.

The MS-DOS and WOW subsystems work differently. MS-DOS programs run in a virtual DOS machine (VDM), an application whose virtual address space contains the code and drivers for MS-DOS. Many VDMs can run simultaneously. When the user enters the name of any program to execute, the VDM asks Win32 to create a new process. VDMs can swap their memory to disk, and they load code only as needed. Also, multiple VDMs share much of their emulation code.

WOW applications also run in a virtual DOS machine. When the user starts a WOW program, Win32 recognizes the DOS executable file format and initiates a VDM. The VDM in turn loads the WOW support into its own address space. All concurrent

WOW applications effectively share a single VDM, just as they would normally share a single DOS machine. The WOW subsystem mimics exactly the old Windows environment.

6. Summary

Windows NT offers true multitasking, multithreading and multiprocessing and extensive networking capabilities. It will run application programs written for a variety of environments, including DOS and Windows, as well as its 32 bit competition, UNIX and OS/2. Windows NT will run on the Intel processor as well as RISC-based PCs and workstations using MIPS 4000 and Digital Equipment's Alpha processors (Feldman, 1993). It also complies with government standards for a C2 security rating (Myers, et al, 1993).

Although Windows NT has many impressive features, its size and complexity are so great that PCs, which started out as liberating users from the control and complexities of mainframe computer centers, have come full circle. The implication for management is big and talented staffs are required to install and maintain this complexity. Instead of the desktop becoming simpler to use, in some respects, it is becoming more difficult.

Microsoft is working on a "lite" version of NT, known variously as Windows 4.0, Chicago, or Windows95. This operating system is purported to have most of the features of Windows NT except its ability to act as a server on client/server networks and its support for OS/2 and POSIX applications. The new operating system's memory and disk requirements will probably be significantly more modest than NT.

IV. GRAPHICAL USER INTERFACES

A. ABOUT USER INTERFACES

The purpose of a user interface is to provide communication between the user, the computer system, and the various application programs. Software design addresses two key questions with regard to user interfaces:

- How does the computer show the user what choices are available?
- How does the user tell the computer what choice has been made?

1. Conventional Interfaces

Conventional interfaces are based solely on text. The reason is that through the 1980's, DOS was strictly text-based, command driven software. Many computers of that time could not handle graphics well. There are four basic types of conventional interfaces discussed here.

Command-line interfaces rely on command typed in by the user. The screen shows a prompt, such as C>, indicating that the computer is ready to accept a command. The user must know the exact syntax for each command, and enter it on the keyboard, one character at a time. The user must also be familiar with the rules for constructing the command, otherwise an error message will be displayed. MS-DOS is an operating system that uses a command-line interface.

Menu-initial interfaces present the user with a series of single letter abbreviations that stand for possible choices. In a menu-initial display, the user must know which letter represents which choice. Often choosing one letter from the menu produces a second-level menu with more letters prompting further choices. Programs that used menu-initial interfaces include the early versions of VisiCalc and SuperCalc.

Menu-word interfaces give the user a list of words instead of initials from which to select. Again, there may be second-level menus with more words indicating more choices. Menu-word interfaces are sometimes called moving bar interfaces. Programs using menu-word interfaces include the MS-DOS versions of Microsoft Word and Lotus 1-2-3.

Menu-driven interfaces present the user with a complete menu from which to choose. The menu takes up the full screen. With each choice, a new menu appears, until all necessary choices have been made. Menu-driven interfaces are good for the novice because they can display full instructions, but they can be frustrating for more experienced users.

2. Shell Interfaces

The software that provides a graphical interface to a command-driven interface is called a *shell*. In actuality, a shell is another layer of software between the user and the command-driven interface (Long, 1993). Shells provide an alternative to often cryptic text commands. With a shell, the user can interact with DOS and other software packages by selecting options from menus that are temporarily superimposed over whatever is currently on the screen or by using a mouse to position the graphics cursor over the appropriate icon. Most shells are text and command based, although a mouse can be used for some operations.

The MS-DOS Shell presents a visual method of working with MS-DOS. The user can perform many of the same file management and disk maintenance tasks that can be performed from the command line. The MS-DOS Shell can also be used to organize and start programs, and to switch between them (Microsoft, 1992).

The WordPerfect Office Shell is a graphical interface which works with the suite of programs included in the WordPerfect Office. The paths to various programs are initially set up and the user selects them from a menu. Hot keys are used to switch between programs within the suite. For non-WordPerfect programs, the user must exit and return to the shell to switch to the next program.

Certain menu programs also provide visual interface with MS-DOS. Examples of this type of shell are DirectAccess and 1DirPlus. The shell used in the Systems Management Laboratories is the 1DirPlus menu program. In this shell, application programs are activated by batch files which have been initially set up by the network administrator. The user is always returned to the shell upon exiting a program in this type of shell.

3. Graphical User Interfaces

Beginning in the 1960's computer research groups began looking for alternatives to the conventional user interface. They identified requirements including ease of use, consistency, and familiarity. Researchers at Stanford Research Institute and Xerox Palo Alto Research Center (PARC) developed an interface incorporating a mouse, icons, and pull-down menus (Lu, 1992). This interface offers many choices without interfering with work in progress. A menu category is chosen from a menu bar and the menu is "pulled down" with a mouse. The work area is only obscured while the user is choosing from the menu.

The PARC interface is found in almost all new microcomputers. Apple built the PARC interface into its Lisa and Macintosh computers. AT&T, Atari and Radio Shack have adopted it. Software such as Microsoft Windows, Geoworks, Ensemble, and Presentation Manager for OS/2 install a PARC interface on IBM compatible PCs. The recent versions of MS-DOS include an optional "shell" that has some features of a PARC interface. (Lu, 1992)

The general acceptance of graphical user interfaces has prompted many software companies to create GUI alternatives for DOS and other popular command-driven software packages. Windows has emerged as the dominant graphical user interface for the MS-DOS environment. It will be discussed in the following section.

B. WINDOWS GUI

Microsoft Windows is not an operating system, but rather a companion program to DOS. It is a graphical user interface which enables users to view applications in various "windows" on their screens. Because Windows is a single-user, single-tasking program, only one program can actually be actively running at any given time. The Windows program switches from one active program to another program giving the appearance that more than one program is active at a time (Schatt, 1992). Under Windows, each application program runs in a window on the screen that can be opened or closed by the user. Multiple applications can be run at the same time, each in its own window, and a user can switch between applications. As a practical matter, multiple windows on the screen simultaneously is limited because the windows become so small that the user cannot effectively use multiple Windows. The task switching is most effective when each application occupies a full screen or half screen at most. However, application programs can be open and minimized, giving the full screen to the program presently active, and still allowing for task switching. The user's knowledge of one application transfers to each new application because all Windows applications operate in a similar fashion.

Windows requires DOS in order to operate (Day, et al, 1993). It is by far the most popular extension and environment for MS-DOS users. It is a true extension because it uses MS-DOS when it must and provides its own services to programs for many interfaces to the hardware.

The addition of Windows to DOS increases hardware requirements significantly. Unlike running DOS alone, DOS with Windows requires a 80386SX processor as a minimum and four megabytes of RAM are recommended (Day, et al, 1992). Windows actually needs eight to 12 megabytes of RAM to run multiple sessions effectively. DOS with Windows together requires approximately 15 megabytes of fixed disk space. Also, the Windows graphical interface overhead for application programs has much greater demand for fixed disk space than DOS applications.

The addition of Windows to DOS also incurs performance penalties. Because Windows uses *timeslice multitasking*, when multiple programs are executed simultaneously, each program gets a section of processing time. As the number of concurrent programs increases, the amount of processing time available for each program decreases. This, in turn, decreases the execution speed for each program (Jasma, 1987). A management concern which must be addressed is the upgrade of existing "good" hardware, such as the 80486/33 processors in the Systems Management Laboratories, to attain efficient performance in the running of Windows. For example, this could be accomplished by replacing existing 33MHz chips with 66MHz chips. These technical and management issues become particularly evident with network operation of Windows. The network aspects will be explored in the following chapter.

Windows supports two types of application programs, DOS and Windows. The user can switch between the Windows environment and the DOS environment to run programs directly in a DOS mode.

1. Multitasking Under Windows

Windows applications share the processor in a *non-preemptive manner*. Windows programs are written to yield control of the processor at regular intervals without user intervention. Programs written for Windows follow a set of guidelines that enables Windows programs to share the processor's time cooperatively. Many applications can be running at the same time, with all of them going through a loop that polls for messages. This scheme depends on well-behaved application programs

The Windows programs are designed to react to messages passed to them from Windows. A keystroke entered is detected by the Windows kernel and the information is passed to the application that has the focus. The concept of *focus* means the application in which the user is working. The user indicates focus by clicking that application's window with the mouse.

Windows can be run in one of two modes, Standard or 386 Enhanced. Each mode has different operating capabilities and uses memory differently. Standard mode operates in extended memory; 386 Enhanced mode operates in extended memory and accesses the virtual memory functions built into the 386 processor. (Person, 1994).

Windows starts in the 386 Enhanced mode if installed on a 386 or 486 PC with more than 2M of memory. When Windows is running in 386 Enhanced mode, it provides *preemptive multitasking* for non-windows applications. Each session is assigned a percentage of the processor's time.

The 386 Enhanced mode creates a "virtual" 8086 on which a non-Windows program can run. Thus the non-Windows program is interacting with the Windows version of MS-DOS and a virtualized computer. When an I/O is requested by a DOS session the Windows version of MS-DOS intercepts the I/O and simulates an 8086 DOS environment. However, many DOS applications will not run, or run inefficiently, in Enhanced Mode. DOS applications should be run under DOS to achieve best performance and greatest efficiency.

2. The Windows Environment

Windows uses concepts that make computers easier for many people to use. The basic organizational concept is that all applications run on a *desktop*, and each application runs in its own window. Windows can run multiple applications. The user can move the windows and change their size just as stacks of papers on a desktop can be moved and rearranged.

Windows has two basic areas in which to perform system commands: the Program Manager and the File Manager (Day, et al, 1993). The Program Manager control window is the starting point for running many utility functions. Within the Program Manager, programs can be started in one of five ways (Day, et al, 1993):

- ◆ Double click the program icon
- ◆ Use the File-Run command and type the program name

- ◆ Highlight the program icon and press Enter
- ◆ Double click the program name under the File Manager
- ◆ Double click a data file that has an association to an executable program

The File Manager can be started from the Program Manager control window. The File Manager performs functions similar to the DOS Shell. Multiple File Manager windows can be opened to display different drives or directories. A toolbar for file and disk management contains tools for tasks such as copying and deleting files, and sharing drives on the network (Person, 1994).

Within Windows are two types of windows: application windows and document windows (Person, 1994). An *application window* contains the application itself. The menu bar that controls the application is usually at the top of the application window, under the application's title. Depending on the type of processor and amount of memory in the computer, multiple applications can be run at one time in Windows. Each application appears in its own application window.

A *document window* is located in an application window. Document windows contain the data or document on which the application works. Some applications, such as Windows Write, can have only one document window open at a time. With other applications, such as Microsoft Excel, multiple document windows can be open at one time.

Windows enables the user to cut or copy information from one application and paste the information into another. With some applications, information can be shared through *linking*. For example, a table of data from a Microsoft Excel worksheet can be linked into a Microsoft Word for Windows document. Then, if changes are made to the original data in the Excel worksheet, the linked data is automatically updated in the Word Document.

Making entries, edits, and changes to information is similar in all Windows applications. The basic procedure is:

- ◆ Activate the window containing the desired application
- ◆ Select the text or graphics object to be changed
- ◆ Choose a command from the menu at the top of the application
- ◆ Select options from a dialog box, if one appears, and complete the command by choosing OK or pressing ENTER

With the mouse, Windows applications can be controlled by choosing commands, selecting options, and moving on-screen items.

3. Object Linking and Embedding

Object linking and embedding (OLE) is a feature that was introduced with Windows 3.1 in 1992 (Person, 1994). An *object* is anything such as data, a picture, or a chart which may become associated with a different application other than the application used to create it. *Linking* is including an object within a different application but leaving the object linked to the application used to create it. In this way the linked object is updated when the original changes. *Embedding* puts data from a server application within the document of a client application. This leaves all the data in the final document, not linked back to its original application. Through OLE, compound documents can be created made up of objects created by many applications.

4. Networking With Windows

a. Peer to Peer Networking

Microsoft developed Windows for Workgroups to allow users who work together to network their workstations. Windows for Workgroups includes enhancements that give it networking capability and features to help people work together. The networking overhead is shared between all the computers on the network. Each user is responsible for their own password, and each user controls access to their own files and

printer. The Windows for Workgroups File Manager and Print Manager allow sharing of directories and printers with others in the workgroup. Additionally, five applications are included which augment workgroup operations:

- ◆ *Mail* is a message-passing system.
- ◆ *Remote Access* gives the ability to log on to a Windows NT server over the telephone.
- ◆ *PC Fax* enables the workgroup members to share a common fax.
- ◆ *Schedule+* is a personal or group scheduling application which ties into the Mail system to notify and confirm meeting times.
- ◆ *Clipbook* enables the members of the workgroup to share data by acting as a central scrapbook for the group. Data can be pasted or linked from the Clipbook into documents so that everyone gets all the updates.

b. Client/Server Networking

Windows can be used on a local area network with a client/server architecture. The selection for configuration of the network is made at the time of installation. There are three ways Windows can be run in a network environment:

- ◆ All Windows files on user workstation
- ◆ All Windows files on server
- ◆ Windows on server/user files on user workstation

Since this is the main issue being addressed in this research, each of these methods will be discussed in great detail in the next section on Graphical User Interface Configurations.

5. Summary

The popularity of Windows is based on users' perceptions of the benefits of graphical interface, multitasking and connectivity between applications. Windows and Windows applications are easier for many to use than DOS and DOS applications, and they are very powerful. Additionally, many Windows applications are customizable and features can be tailored to meet the specific needs of the user.

Windows 3.1 was released in the spring of 1992. The current version, Windows 3.11, was released in the spring of 1994 (Person, 1994). The network features are the newest addition to Windows 3.0 and 3.1.

The next upgrade to Windows will be Windows 4.0, also called Windows Chicago or Windows95. This will be a 32 bit, self contained, DOS independent operating system. This is advertised to be a "lite" version of Windows NT and will have most of the features of Windows NT except its ability to act as a server on a client/server network and its support for OS/2 and POSIX applications (Feldman, 1993).

V. GRAPHICAL USER INTERFACE CONFIGURATIONS

A. CONFIGURING WINDOWS ON A LAN

Configuration of Windows on a local area network involves several important issues. Careful consideration should be given to the proper balance of usability, performance, ease of maintenance and administration. The configuration question is whether Windows should be placed on the file server or on the local hard disk of the user computer. Different networking environments require different configurations. PCs, the network board, cabling and printers comprise the physical network environment. Additionally, needs and proficiencies of users on the network must be identified. The following requirements must be studied:

- ◆ *User Flexibility.* This refers to the flexibility the users are allowed over their own Windows configuration.
- ◆ *Ease of Administration.* This is the degree of difficulty of network administration which will allow for effective management and maintenance of the network.
- ◆ *Performance.* The different ways of configuring Windows affects performance for the user and for the network at large.
- ◆ *Reliability.* The different ways of configuring Windows affects reliability for the user and for the network at large.

In addition to the question of how to configure Windows, a location must be determined to put virtual memory swap files and temporary files. These files do not necessarily reside in the same location as Windows. Since virtual memory swap files and temporary files can affect performance, their location is another question to be addressed when deciding on how to configure Windows.

B. ALL WINDOWS FILES ON USER COMPUTER

This configuration consists of a working copy of Windows 3.1x on the user's local drive. Windows program files and user files reside on the user computer hard drive. The network administrator has little control over the operation of a user computer with this configuration.

1. Files on User's Disk

Full installation to the user's local drive creates two directories, the target directory where the Windows program files will reside and the SYSTEM subdirectory which is the user files. The default name for the target directory is WINDOWS, which contains approximately 125 files that use about 4.5M of disk space. This directory contains Windows bundled programs such as Calendar, Notepad, Cardfile and the related help and DLL files. It also contains bit-map wallpaper files, screen-saver files, sound files, WindowsINI files, PIF files, group GRP files, and the program file WIN.COM.

The SYSTEM subdirectory contains about 115 files which use about 5.5M of disk space. The Windows core executable files and their associated DLL files are located here. Also in this subdirectory are the TrueType font files, the fixed screen fonts, hardware drivers, screen grabbers and the network drivers. (Person, 1994)

2. The Systems Management Laboratories

The Systems Management Laboratory in I-250 has workstations that were acquired with Windows already installed. Thus it was simple to use this configuration. To operate Windows in this configuration, the user selects an application batch file which starts Windows. In the Systems Management Laboratories this is the method used to access all applications, whether stored on the server or on the user computers. In this case the user's local C: drive is the target directory for the Windows program files as well as the location of the user files.

This method of operation leaves all Windows files on the user's local disk, but allows for network access of application programs on the server.

a. Standard Screen

In order to get each machine on the network to start Windows with a standard Program Manager screen, several steps were performed:

- The Standard Screen was configured on one user computer with the Control Panel of the Main Group removed from the icon set to preclude a user changing colors, fonts, etc. on the Program Manager screen.
- A Restrictions Section was added to the PROGMAN.INI file to disallow the user from changing the setup of the Program Manager by adding the settings EditLevel=4 and NoSaveSettings=1. The restrictions prevent users from adding, deleting and copying icons and disable the option to save screen settings.
- The GRP and INI files representing the standard screen were then placed in a STANDARD.SET directory on the network I: drive.
- The settings were copied down to each user computer from the server with the command XCOPY I:\WINDOWS\STANDARD.SET C:\WINDOWS

b. Windows Load Time

SMARTDrive is a disk cache that comes with Windows. A *disk cache* works by reducing the amount of writing and reading of information to and from the hard disk. SMARTDrive sets aside an area of extended memory as a cache of disk information for Windows. Because the cache is electronic memory, retrieval can be hundreds or even thousands of times faster than a mechanical disk drive (Person, 1994).

In order to maximize performance, the statement SMARTDRV.EXE was added to the batch file which initializes Windows. The result was a load time of 18 seconds.

3. Advantages

a. No Usage of File Server Disk Space

All Windows files are installed on the user's PC.

b. Reduced Network Traffic

Installation of Windows on the local disk instead of the file server keeps Windows traffic off the network. Windows also uses many different dynamic link libraries (DLLs). Installation of Windows locally reduces the load associated with loading the Windows DLLs.

c. Improved Windows Load Time

The local hardware platforms run faster than the combination of the file server hardware and the LAN cabling system, so loading Windows from a local disk reduces the amount of time it takes to load Windows. Tests showed that Windows could be loaded in an average of 18 seconds. Tables 1, 2 and 3 show results of timing tests performed.

d. Windows Hardware Configuration Files Match the Machine

The SYSTEM.INI file is a hardware specific configuration file. With a local installation, the SYSTEM.INI file reflects the settings needed for the particular user computer.

4. Disadvantages

a. Large Amount of Disk Space is Required

Windows 3.1x consumes 9-14M of user's disk space depending upon the installation options chosen.

b. Administration of Workstations is Decentralized

Management of the users' environment by the network administrator is impossible without actually physically accessing each user's user computer.

C. ALL WINDOWS FILES ON SERVER

This variation of the Windows 3.1x configuration has all information on the file server. All shared Windows program files are installed on the file server in a shared directory, and all user specific files are installed into a private user directory on the file server.

1. Files on User's Directory

The files on the user's private directory on the file server are the same as those installed on the user's disk in the previous configuration. The difference is the target directory, which is on the file server with this method and was on the user's local drive with the previous method.

2. The Systems Management Laboratories

The Systems Management Laboratories do not have a configuration setup for this method of loading Windows. Some experimental setups were conducted to test this configuration method. The INSTALL function was run using the network J: drive as the Windows program directory and the network K: drive as the location for the user files. The INSTALL was run by typing SETUP/N at the J:> prompt. The switch /N indicates network setup. The INSTALL was successful, but the program would not load properly. Error message "NOT READY READING DRIVE K" appeared three times during the loading of Windows. The program did finally load, but the cause of the read failure could not be located.

Additionally, when installed in this configuration on two machines, Windows could not be accessed simultaneously by both users. A sharing violation prevented one user from loading the program. This indicates the need for a separate user directory for each user computer if this method were used.

This configuration is not efficient in terms of performance or use of resources. It is usually only recommended for diskless workstations or workstations that must be booted from a floppy disk.

3. Advantages

a. No Usage of Local Disk Space

This method places all Windows program files and user files on the file server, so no disk space is used on the user computer.

b. Centralized Updating of Windows Files and Drivers

Installation of updated versions of the files is easier because all core Windows files reside on the file server.

c. Centralized Administration is Possible

The network administrator has complete control over the user's environment since all files needed to run Windows reside on the file server. This allows for easier administration and troubleshooting on the network by the network administrator.

4. Disadvantages

a. Large Amount of File Server Disk Space is Required

Windows 3.1x consumes approximately 15M of disk space on the file server. This configuration also requires a separate user directory for each user or machine.

b. Network Traffic for Windows is Increased

Since all files are on the file server, DLLs, swap files and temporary files also reside on the file server. This will greatly increase traffic demands on the network.

D. WINDOWS ON SERVER/USER FILES ON USER COMPUTER

This technique of configuring Windows places a small group of user files on the user's local drive. These are the WindowsINI files, the GRP files, the initial PIF files, and WIN.COM. This mode dictates that the major program files needed to run Windows be located in a shared directory on the file server. The user needs a search path from the user computer to the Windows directory on the file server in order to run Windows from the user computer.

This method is ideal for centralized network administration. Also this approach minimizes the amount of disk space needed on the local drive.

1. Files on User's Disk

The following user files are installed by Windows on the user's local drive:

WIN.INI	MOUSE.INI
CONTROL.INI	WINVER.EXE
DOSPRMPT.PIF	WIN.COM
PROGMAN.INI	MAIN.GRP
ACCESSOR.GRP	GAMES.GRP
STARTUP.GRP	SYSTEM.INI
<u>_DEFAULT.PIF</u>	

The file WIN.COM is not part of Windows, instead it is a program that launches Windows. Setup creates WIN.COM and this program is explicit to the hardware specified during the Setup program. (Person, 1994)

2. The Systems Management Laboratories

The Systems Management Laboratory in I-250 has the Windows files installed in this configuration for the purpose of the network administrator experimenting with this configuration. This installation was performed using the SETUP/N option. To operate Windows in this mode, the user selects a batch file which contains a path back to the user's C: drive for the user files and to the network J: drive for the Windows program files. The batch file also contains a command which copies down a standard set of INI files each time the Windows program is loaded. In this way, the initial Windows screen should be standard for all network users.

a. File Setup

Each user machine must have the user files installed on the local drive to run the Windows program from the file server. This could be accomplished in one of two ways. The first method consisted of running the INSTALL program at the user machine from the J:\WINDOWS file. This procedure consumed approximately four and a half minutes at each user machine.

The second method was to simply copy the set of user files from a network drive to each user computer. The required files were loaded to the K: drive, and then at each machine the following commands were entered: C:>MD WINUSER to make a user directory to receive the files, and then C:>XCOPY K:\WINUSER to copy down the files. This procedure took approximately 30 seconds at each machine, with an actual file copy time of about two seconds.

The user access to Windows is provided through a batch file. The network user boots up directly into a 1DirPlus Menu screen which accesses program batch files on the network F: drive. A maintenance batch file called ZZWIN.BAT was written for these experiments. The batch file directs the program to J:\WINDOWS on the network drive for the Windows program and C:\WINUSER for the user files needed to run Windows. From the moment of selection of the ZZWIN batch file by the user, the Windows program took about 31 seconds to load.

b. Standard Screen

In order to get each machine on the network to start Windows with a standard Program Manager screen, the INI files in a STANDARD.SET directory are copied down to the C: drive from the network J: drive. The STANDARD.SET is a directory which was set up on the J: drive specifically for this purpose containing all INI files required to start Windows. Also, restrictions were set on the Program Manager screen so that users could not add, delete or copy program icons and the option to save screen settings was disabled. The restrictions were enacted by changing settings within the PROGMAN.INI file. This was accomplished as follows:

- ♦ Within the Group Section of the PROGMAN.INI file, the paths were changed to J:\WINDOWS\STANDARD.SET*.GRP to get the group files from the server.
- ♦ A Restrictions Section was added to the PROGMAN.INI file to disallow the user from changing the setup of the Program Manager by adding the settings EditLevel=4 and NoSaveSettings=1.
- ♦ The batch file on F: was changed to read:
XCOPY J:\WINDOWS\STANDARD.SET*.INI C:\WINUSER

c. Windows Load Time

The Windows load time initially tested at over two and a half minutes using the server/user configuration without SMARTDrive. Local hard disk activity seemed to be consuming most of the load time. After adding the statement SMARTDRV.EXE to the batch file which initializes Windows, load time was reduced from 2 minutes 33 seconds to 31 seconds.

3. Advantages

a. Limited Usage of Local Disk Space

This method places the majority of the Windows files on the file server. The small group of user files needed locally requires 150K or less of disk space.

b. Centralized Updating of Windows Files and Drivers

Because all core Windows program files reside on the file server, updated versions of these files will be easier for the network administrator to install.

c. Windows Configuration Files Match the Machine

The hardware specific SYSTEM.INI configuration file resides on the local drive. By being installed locally, the SYSTEM.INI file will reflect the settings needed for that user computer.

4. Disadvantages

a. Increased Network Traffic Caused by Windows

Any application loaded over the network will increase traffic. Windows is a large application that can generate significant traffic, which will impact network throughput.

b. Administration of User's PC is Mixed

The user's environment cannot be totally managed by the network administrator without physically accessing each user's machine, since some of the user's Windows files will be installed locally. This disadvantage has been minimized to a great extent

in the Systems Management Network. The network administrator has gained a degree of control of the users' environments with the batch file which replaces all INI files each time a user loads the Windows program, and the restrictions placed on the Program Manager screen to preclude changes.

c. Degraded Windows Load Time

This is a disadvantage if the file server and network hardware perform slower than the local drive. In the case of the Systems Management Laboratories the file server hardware is identical to the user computer hardware, thus the file server is slower due to the addition of network processing time for the files. Tests showed that Windows could be loaded in an average of 31 seconds from the network drive, compared to 18 seconds from the local drive.

E. PERFORMANCE TEST RESULTS

Comparison of the various methods of configuration of Windows on the network included testing of the different configurations for performance. Timing tests were performed on different user machines and with a group of machines simultaneously. Two configurations were tested and compared: all Windows files on the user computer, referred to in the tables as *Windows (User)*; and Windows files on the server with user files on the user computer, referred to as *Windows (Server)*.

Access times were tested for loading Windows as well as loading applications. The typical applications used in the test were the following versions of WordPerfect: Version 5.1 for DOS; Version 6.0 for DOS; and WordPerfect for Windows (WPW) Version 6.0a for Windows.

The performance results tables are presented as follows: the first row is access time for WordPerfect (DOS) 5.1; the second row is access time for WordPerfect (DOS) 6.0; the third row is access times for Windows (User) and WordPerfect for Windows; and the fourth row is access times for Windows (Server) and WordPerfect for Windows.

1. One 33MHz User Computer

Table 1 shows the results of timing tests performed on one 33MHz 486 IBM compatible user computer on the Token Ring Network OTR in the Systems Management Laboratory.

Time To:	Windows (User)	Windows (Server)	WP(DOS) v5.1	WP(DOS) v6.0	WPW v6.0a	TOTAL TIME
F:>	*	*	11 sec	*	*	11 sec
F:>	*	*	*	22 sec	*	22 sec
F:>	18 sec	*	*	*	27 sec	45 sec
F:>	*	31 sec	*	*	29 sec	60 sec

Table 1. Access Times for One 33MHz Machine

2. One 66MHz User Computer

Table 2 shows the results of timing tests performed on one 66MHz 486 IBM compatible user computer on the Token Ring Network OTR in the Systems Management Laboratory.

Time To:	Windows (User)	Windows (Server)	WP(DOS) v5.1	WP(DOS) v6.0	WPW v6.0a	TOTAL TIME
F:>	*	*	10 sec	*	*	10 sec
F:>	*	*	*	22 sec	*	22 sec
F:>	14 sec	*	*	*	21 sec	35 sec
F:>	*	30 sec	*	*	27 sec	57 sec

Table 2. Access Times for One 66MHz Machine

3. Five 33MHz Workstations

Table 3 shows the results of timing tests performed on five 33MHz 486 IBM compatible workstations simultaneously on the Token Ring Network in the Systems Management Laboratory.

Time To:	Windows (User)	Windows (Server)	WP(DOS) v5.1	WP(DOS) v6.0	WPW v6.0a	TOTAL TIME
F:>	*	*	24 sec	*	*	24 sec
F:>	*	*	*	49 sec	*	49 sec
F:>	21 sec	*	*	*	43 sec	64 sec
F:>	*	55 sec	*	*	87 sec	142 sec

Table 3. Access Times for Five 33MHz Machines

F. SUMMARY

The decision to configure Windows should be made by weighing the following factors:

1. Best Performance

If the objective is to install Windows for the fastest performance, even at the expense of more work for the network administrator, Windows should be installed on the user hard drive. This minimizes traffic on the network. By adding more memory to the current 8M of RAM installed on the user machine, there will be no need for a virtual memory swap file and there will be enough memory for a 4-8M RAM disk for temporary files. Figure 1 shows a representation of this configuration.

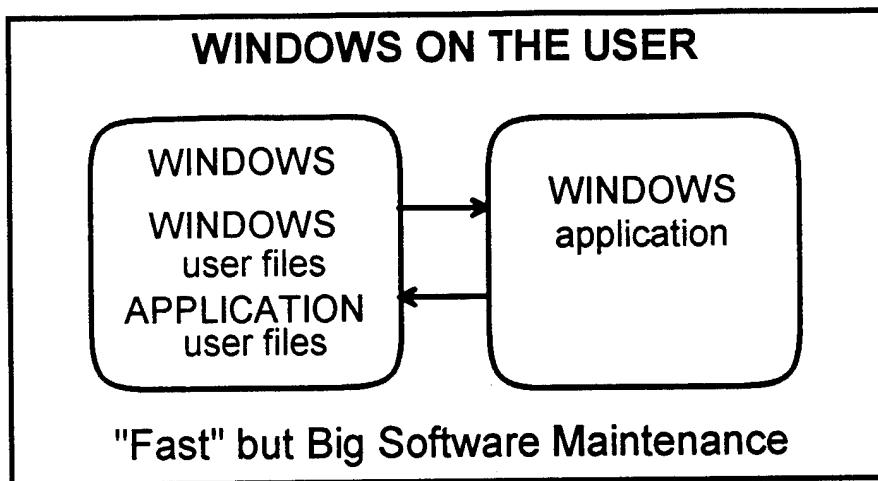


Figure 1. Windows on the User. From Schneidewind, 1995

2. Best Administration

The best administrative control will be achieved with a network installation, keeping all Windows core files on the file server and the user-specific files in the user's local directory. This will allow the network administrator to easily make changes as needed to Windows files.

This method would result in slower Windows loading caused by accessing Windows from the file server. Figure 2 shows a representation of this configuration.

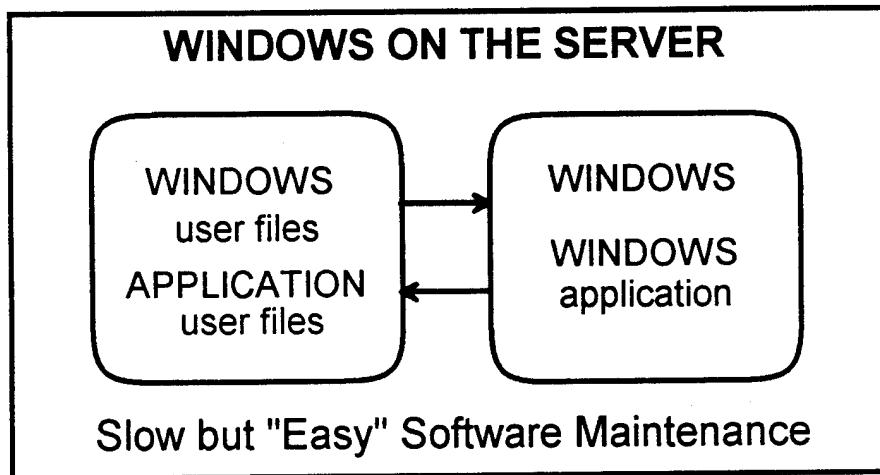


Figure 2. Windows on the Server. From Schneidewind, 1995

VI. CONCLUSIONS AND RECOMMENDATIONS

A. PERFORMANCE IMPLICATIONS

The decision to configure Windows must be made with consideration to compatibility and interoperability, as well as performance and administration. Windows and applications on the network must be easily accessed by students and faculty in support of academic endeavors, as well as robust enough to be used in a classroom setting. Performance should be supportive of many users accessing the network without significant degradation in response times. Administration of the network should be possible within constraints of available monetary and personnel resources.

The absolute fastest performance would be achieved with all Windows and Windows applications on the user's local drive. Clearly this is not practical, and in fact obviates the need for a client/server network.

Conversely, performance times with a network installation of Windows are unacceptably slow. The time differences between one machine and five machines accessing Windows and an application on the network is more than double the time. If the laboratory were fully utilized, as would happen in a classroom environment, twenty machines would need to access Windows and an application on the network simultaneously.

Not only would performance times be severely degraded, but the files needed by the server would not be sufficient to support such an operation. Since the PC LAN Program network operating system runs over DOS, it is subject to the files limitations of DOS. The maximum files that can be in use on the server is 255. With each Windows user utilizing 14 files just for Windows alone, a full classroom would not be able to access a network application after each user opened Windows on the network.

A solution to the configuration question must necessarily compromise between performance of the network, usability by faculty and students, and administration issues.

B. CONFIGURATION RECOMMENDATION

Based on experimentation and performance timing of the various configurations for Windows, the recommendation is submitted that **ALL WINDOWS FILES ON USER** is the optimal setup for the Systems Management laboratories.

This configuration is essentially already implemented in OTR, since each workstation had Windows installed at the time of purchase. Windows on the user drive accessing applications on the network drive will reduce network traffic and improve performance times.

For implementation in the other Systems Management laboratories which do not have Windows, the installation can be performed at every user workstation by using the SETUP/A option on the server. Then each machine can be logged on to the network and can load the Windows files from the server without using the disks for installation every time.

The Windows configuration with all files on each user workstation will allow the flexibility needed for the addition of application programs to the server which may or may not operate under Windows. All software applications which may be required by students and professors in the future will be able to be integrated into the network.

LIST OF REFERENCES

Abernathy, Joe, "This is Not Your Father's OS/2," *PC World*, vol. 12, no. 8, p. 72, August 1994

Bertolucci, Jeff, "DOS-of-the-Month Club: IBM PC DOS 6.2," *PC World*, vol. 12, no. 7, p. 60, July 1994

Day, Michael, Larry Budnick, Eric Harper, Blaine Homer, John Jerney, Jim Krochmal, Rita Lewis, and Dennis Williams, *LAN Operating Systems*, New Riders Publishing, 1993

Deitel, H. M., *Operating Systems*, Addison Wesley Publishing Company, 1990

Delfino, Erik, "Windows Multitasking," *Database*, vol. 16, no. 6, p. 87, December 1993

Feldman, Len, "Windows NT," *Compute*, vol. 15, no. 8, p. 5, August 1993

Long, Larry and Nancy, *Computers*, Prentice Hall, 1992

Lu, Cary, *The Apple Macintosh Book*, Microsoft Press, 1992

Markoff, John, "Microsoft to Introduce New Windows NT," *The New York Times*, vol. 143, p. 19, September 17, 1994

Mendelson, Edward, "IMB's PC DOS 6.1: Has MS-DOS Met Its Match?" *PC Magazine*, vol. 12, no. 16, September 28, 1993

Mendelson, Edward, "And Novell Makes Three: Is This the Right DOS for You?" *PC Magazine*, vol. 13, no. 7, April 12, 1994

Microsoft, *Microsoft MS-DOS Operating System 5.0 User's Guide and Reference*, Microsoft Corporation, 1992

Minasi, Mark, "DOS 6.0: DOS Reaches New Levels," *Compute*, vol. 15, no. 6, June 1993

Minasi, Mark, "MS-DOS 6," *Compute*, vol. 15, no. 9, September 1993

Myers, Brian and Eric Hamer, *Mastering Windows NT Programming*, Sybex, 1993

NPS, *Administrative Science Department Proposal for Local Area Network Installation*, Administrative Science Department, Naval Postgraduate School, Monterey, CA, 1985

Person, Ron, *Using Windows, 3.11 Edition*, Que Corporation, 1994

Phaniraj, V., *OS/2 New User Introduction*, North Dakota State University, 1994

Roberts, Tony, "Old DOS, New Tricks," *Compute*, vol. 16, no. 3, March 1994

Schatt, Stan, *Understanding Local Area Networks*, Sams, 1992

Schneidewind, Norman F., "Principles of Local Area Networks," *Encyclopedia of Science and Technology*, McGraw-Hill, 1986

Schneidewind, Norman F., *IS3503 Class Notes*, Naval Postgraduate School, Monterey, CA, 1995

Tannenbaum, Andrew, *Modern Operating Systems*, Prentice Hall, 1992

Wolverton, Van., "DOS vs. DOS vs. DOS," *PC World*, vol. 12, no. 1, January 1994

INITIAL DISTRIBUTION LIST

1.	Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2.	Library, Code 52 Naval Postgraduate School Monterey, CA 93943-5101	2
3.	Norman F. Schneidewind, Code SM/Ss Naval Postgraduate School Monterey, CA 93943-5101	1
4.	William B. Short, Code SM/Sh Naval Postgraduate School Monterey, CA 93943-5101	1
5.	M. J. Lanni 159 Suncliff Place Vallejo, CA 94591	1